providing a first instruction causing loading of a first part of said unaligned data into a first storage location by using a first pointer giving a memory address of a first position;

rotating and masking and sign-extending said first part of said unaligned data in said first storage location from the first position to a second position;

providing a second instruction causing loading of a second part of said unaligned data into a second storage location by using a second pointer giving a memory address of a fourth position;

rotating and masking said second part of said unaligned data in said second storage location from a third position to the fourth position; and

providing a third instruction causing combining of said first storage location with said second location using a logical operation into a result storage location.

2. The method of claim 1 wherein said first pointer is a first register, said first storage location is a second register, said second pointer is a third register, and said second storage location is a fourth register, and said result storage location is a result register.

3. The method of claim 2 wherein said registers are 64-bits in length.

4. The method of claim 1 wherein the logical operation is a bit-wise OR operation.

6. The method of claim 1 wherein said rotating is performed in two phases comprising a first phase in which a major rotation is performed and a second phase in which a minor rotation is performed.

7. (Twice amended) A method for storing data into an unaligned plurality of memory locations, comprising:

providing a first instruction causing rotation of data in a first storage location and sign-extending and storing of a first part of said data in a first portion of unaligned plurality of memory locations from a first position to a second position;

having a first pointer giving an address of the first position;

providing a second instruction causing rotation of data in a second storage location and storing of a second part of said data in a second portion of unaligned plurality of memory locations from a third position to a fourth position; and

having a second pointer giving an address of the fourth position.

8. The method of claim 7 wherein said first pointer comprises a high address and said second pointer comprises a low address.

9. The method of claim 8 wherein said data is stored in said unaligned plurality of memory locations inclusively between said high address and said low address.

10. The method of claim 7 wherein said rotating is performed in two phases comprising a first phase in which a major rotation is performed and a second phase in which a minor rotation is performed.

11. The method of claim 7 wherein said first storage location is a first register, said first pointer is a second register, and said second pointer is a third register.

12. The method of claim 7 wherein said data is selected from a group consisting of data 16, 32, and 64 bits in length.

13. The method of claim 3 wherein said unaligned data is 16 or 32 bits in length, and said first instruction further comprises sign-extension when said unaligned data is in big endian order and said second instruction further comprises sign-extension when said data is in little endian order.

14. A method for loading unaligned data stored in a plurality of memory locations, comprising:

loading a first part of said unaligned data into a first storage location;

rotating and sign-extending said first part of said unaligned data in said first storage location from a first position to a second position;

loading a second part of said unaligned data into a second storage location;

rotating said second part of said unaligned data in said second storage location from a third position to a fourth position; and

combining said first storage location with said second location using a logical operation into a result storage location.

15. The method of claim 14 wherein said first storage location is a first register, said second storage location is a second register, and said result storage location is a result register.

16. The method of claim 15 wherein said registers are 64-bits in length.

17. The method of claim 14 wherein the logical operation is a bit-wise OR operation.

18. The method of claim 14 wherein said rotating is performed in two phases comprising a first phase in which a major rotation is performed and a second phase in which a minor rotation is performed.

19. A method for storing data into an unaligned plurality of memory locations, comprising:

rotating and sign-extending a first part of said data in a first storage location from a first position to a second position;

storing said data located in second position in said unaligned plurality of memory locations at an address given by a first pointer;

rotating a second part of said data in a second storage location from a third position to a fourth position; and

storing said data located in forth position in said unaligned plurality of memory locations at an address given by a second pointer.

20. The method of claim 19 wherein said first storage location is a first 64-bit register, said second storage location is a 64-bit second register, and said result storage location is a 64-bit result register.

21. The method of claim 19 wherein said data is selected from a group consisting of data 16, 32, and 64 bits in length.